



HTML の学習

Hyper Text Markup Language

— 2009 年度版 —

福岡工業大学
情報工学部 情報工学科
柴田望洋

BohYoh Shibata
Fukuoka Institute of Technology

■ 本資料について

- 本資料は、2009年度・福岡工業大学 情報工学部 情報工学科2年生の講義

『オペレーティングシステム』

の補助テキストとして、福岡工業大学 情報工学部 情報工学科 柴田望洋が編んだものです。
自分で作るテキスト兼ノートとして活用しましょう。

- 参考文献・引用文献等は、資料の最後にまとめて示します。

■ 諸君が本資料をファイルに綴じやすいように、研究室の学生達が時間を割いて、わざわざ穴を開けるという作業を行っています（一度のパンチで開けることのできる枚数は限られており、気の遠くなるような時間がかかっています）。

必ずB5のバインダーを用意して、きちんと綴じましょう。

■ 本資料のプログラムを含むすべての内容は、著作権法上の保護を受けており、著作権者である柴田望洋の許諾を得ることなく、無断で複写・複製をすることは禁じられています。

- 本講義では、以下に示すホームページ上の、各ドキュメントも参照します。

柴田望洋後援会オフィシャルホームページ <http://www.BohYoh.com/>

- 本資料は、Adobe社のDTPソフトウェアInDesign CS4を用いて作成しています。

本資料は、1年生の講義『コンピュータリテラシー』において、HTMLを学習済みであることを前提としています。本資料を読み進めるためには、ある程度HTMLの基礎知識をもっていることが必要です。

■ ホームページ

ホームページのサンプルです。

ホームディレクトリ（ドライブH）のWWWディレクトリの中のOSディレクトリの中にindex.htmlという名前で作成します)

- ▶ WWWディレクトリおよび、その中のOSディレクトリがなければ作成します。

List 1

index.html

```
<HTML>

<HEAD>
<TITLE>
柴田望洋のオペレーティングシステム用ホームページ
</TITLE>
<STYLE TYPE="text/css">
  A {text-decoration: none;}
  A:link {color: blue; text-decoration: none;}
  A:visited {color: blue; text-decoration: none;}
  A:active {color: red; text-decoration: none;}
  A:hover {color: red; background-color: lightgreen; }
  BODY { background-color: black; color: yellow; }
</STYLE>
</HEAD>

<BODY>
ようこそ柴田望洋のオペレーティングシステム用ホームページへ!!<BR>
僕は<A HREF="http://www.fit.ac.jp/">福岡工業大学</A>の学生です。<BR>
情報処理技術者試験の解答は<A HREF="KOTAE.xlsx">ここ</A>です。<BR>
僕の自己紹介のページは<A HREF="introduction.html">こちら</A>です。<BR>
</BODY>
</HTML>
```

ようこそ柴田望洋のオペレーティングシステム用ホームページへ!!
僕は福岡工業大学の学生です。
情報処理技術者試験の解答はここです。
僕の自己紹介のページはこちらです。

■ 自己紹介のページ

自己紹介ページのサンプルです。

ホームディレクトリ（ドライブH）のWWWディレクトリの中のOSディレクトリの中に introduction.html という名前で作成します）

List 2

introduction.html

```
<HTML>

<HEAD>
<TITLE>
柴田望洋の自己紹介ホームページ
</TITLE>
<STYLE TYPE="text/css">
  A {text-decoration: none;}
  A:link {color: yellow; text-decoration: none;}
  A:visited {color: yellow; text-decoration: none;}
  A:active {color: white; text-decoration: none;}
  A:hover {color: white; background-color: lightgreen; }
  BODY { background-color: green; color: white; }
</STYLE>
</HEAD>

<BODY>
柴田望洋の自己紹介のページ
<HR>
僕の名前は、柴田望洋。福岡県の出身です。<BR>
<HR>
<A HREF="index.html">ホームページへ戻る</A>
</BODY>

</HTML>
```

柴田望洋の自己紹介のページ

僕の名前は、柴田望洋。福岡県の出身です。

[ホームページへ戻る](#)

■ 代表的な色

代表的な色を示します。色名を英語の識別子で指定する方法と、16進数(RGBの各値を、それぞれ0～FFの256階調としたものの並び)で指定する方法とがあります。

▶ RGBは、Red/Green/Blueの光三原色です。

色	色を表す識別子	RGB値
黒	black	#000000
ネイビー	navy	#000080
深青	darkblue	#00008B
青	blue	#0000FF
深緑	darkgreen	#006400
緑	green	#008000
深水色	darkcyan	#008B8B
ライム	lime	#00FF00
アクア	aqua	#00FFFF
水色	cyan	#00FFFF
薄い水色	lightcyan	#E0FFFF
紫	purple	#800080
オリーブ	olive	#808000
灰色	gray	#808080
薄い灰色	lightgrey	#D3D3D3
黄緑	yellowgreen	#9ACD32
茶色	brown	#A52A2A
緑黄	greenyellow	#ADFF2F
銀	silver	#C0C0C0
チョコレート	chocolate	#D2691E
赤	red	#FF0000
マゼンタ	magenta	#FF00FF
濃いピンク	darkpink	#FF1493
オレンジ	orange	#FFA500
ピンク	pink	#FFC0CB
トマト	tomato	#FF6347
金色	gold	#FFD700
雪	snow	#FFF0FA
黄色	yellow	#FFFF00
薄い黄色	lightyellow	#FFFFE0
白	white	#FFFFFF

▶ HTMLのトリビア：

灰色はgreyとgrayのどちらでもいいのですが、薄い灰色はlightgreyでなければなりません(Internet Explorerの場合)。

■ カスケーディングスタイルシート（CSS）

本来、HTMLは文書の《構造》を定義するための言語であり、色などの見栄えを定義するための言語ではありません。したがって、《見栄え》は、HTMLではない手段を用いて定義したほうが好ましいのです。

そのために使われるのがカスケーディングスタイルシートです（以下、単に“スタイルシート”と呼ぶことにします）。

■スタイルシート定義

スタイルシートの定義は、以下の形式で行います。

```
<STYLE TYPE="text/css">
```

```
... 定義 ...
```

```
</STYLE>
```

ただし、スタイルシートに対応していないブラウザを考慮して、以下のように宣言するのが一般的です。

```
<STYLE TYPE="text/css">
```

```
<!--
```

```
... 定義 ...
```

```
-->
```

```
</STYLE>
```

<!-- と --> で囲まれた部分は、HTMLではコメント（注釈）とみなされます。そのため、スタイルシートの定義内容は、スタイルシートに対応していないブラウザでは、実質的に無視されることになります。

定義の箇所は、以下の書式で記述します。

```
セレクタ { プロパティ : 値 }
```

■ 文字の表示属性定義

スタイルシートを用いて、文書中の文字の色を設定する方法を以下に示します。

```
<STYLE TYPE="text/css">
<!--
  body { color : white;
        background-color : black
      }
  A {text-decoration: none;}
  A:link {color: blue; text-decoration: none;}
  A:visited {color: blue; text-decoration: none;}
  A:active {color: red; text-decoration: none;}
  A:hover {color: red; text-decoration: none;}
-->
</STYLE>
```

各セレクトアの概要は、次のとおりです。

セレクトア	解説
body	本文の色、背景色などを定義します。
A	本文中のリンク文字属性を定義します。
A:link	リンク文字の属性を定義します。
A:visited	既に訪れたリンク文字の属性を定義します。
A:active	アクティブなリンク文字の属性を定義します。
A:hover	マウスがのせられているリンク文字の属性を定義します。

セレクトア **body** に指定するプロパティの概要は、次のとおりです。

プロパティ	解説
color	本文の文字の色を定義します。
background-color	本文の背景色を定義します。

プロパティ **text-decoration** に指定する値は、以下のようになっています。

セレクトア	解説
none	テキストに線が付かず、点滅もしません。
underline	テキストに下線が付きます。
overline-color	テキストに上線が付きます。
line-through	テキストに打ち消し線が付きます。
blink	テキストが点滅します。

■ 背景画像の指定

ホームページの背景画像は、以下のようにHTMLのBODYタグで設定できるようになっています。

```
<BODY BACKGROUND="画像ファイル名">
... 中略 ...
</BODY>
```

しかし、スタイルシートで指定したほうが、高機能で柔軟な設定が可能です。

↓ 右側の部分は解説です。書いたら駄目です。

```
<STYLE TYPE="text/css">
<!--
body {
  background-image:url("画像ファイル名");

  background-attachment:fixed;      ① 固定する (スクロールしない)。
  background-repeat:no-repeat;     ② 1枚だけ表示する
  background-repeat:repeat-y;      ③ 縦方向のみ繰り返す。
  background-repeat:repeat-x;      ④ 横方向のみ繰り返す

  ※ ②、③、④は、
     どれか一つだけを指定するか、
     まったく指定しないようにすること。

  background-position:99% 99%;    ⑤ 背景画像のX方向、Y方向の位置を指定。
                                   場所の指定99は0~100の値。
                                   たとえば、以下のように②と組み合わせて指定すると
                                   背景画像が1枚だけ (中央) に表示されます。
  background-repeat:fixed;
  background-position:50% 50%;
  画面の (端) に表示したいのであれば、
  場所の指定に0%や100%を使うこと。
}
-->
</STYLE>
```

ホームページ (index.html) と、自己紹介のページ (introduction.html) に、それぞれ別の文字スタイル・背景画像を設定してみましょう。

■ HTML の基本的なタグ

ここでは、HTML の基本的なタグの概略を紹介します。

改行します。

<PRE> … </PRE>

テキスト中の空白や改行に意味を持たせませす (空白や改行が無視されません)。すなわち、“整形済み”であることを指定します。

<HR>

水平線を表示します。

NOSHADE	立体的にしない。
WIDTH = n	長さを n ピクセルにする。
WIDTH = n%	長さをブラウザ表示幅の n% にする。
SIZE = n	太さを n ピクセルにする。
ALIGN = LEFT	左側に揃える。
ALIGN = CENTER	中央に揃える。
ALIGN = RIGHT	右側に揃える。
COLOR = color	色を指定する。

<DIV> … </DIV>

テキストの分割領域の指定をします。

ALIGN = LEFT	分割された部分を左揃えとする。
ALIGN = CENTER	分割された部分を中央揃えとする。
ALIGN = RIGHT	分割された部分を右揃えとする。

 …

フォントのサイズや色などを指定します。

SIZE = n	サイズを n にする ($1 \leq n \leq 7$)。既定値は 3。
SIZE = +n	サイズを n だけ大きくする。
SIZE = -n	サイズを n だけ小さくする。
COLOR = color	色を指定する。

URLで指定された画像を表示します。

ALIGN = TOP	画像の上端が文字と一致するように配置
ALIGN = MIDDLE	画像の中央に文字のベースラインが一致するように配置
ALIGN = BOTTOM	画像の下端に文字のベースラインが一致するように配置
ALIGN = LEFT	画像を左端に配置し、文字が回り込むようにする
ALIGN = RIGHT	画像を右端に配置し、文字が回り込むようにする
ALT = string	画像が表示できない環境・状態などで、画像の代わりとして文字 string を表示する

<MARQUEE> … </MARQUEE>

電光掲示板のような流れる文字を表示します。

WIDTH = n	領域の長さを n ピクセルにする。
WIDTH = n%	領域の長さをブラウザ表示幅の n% にする。
HEIGHT = n	領域の高さを n ピクセルにする。
HEIGHT = n%	領域の高さをブラウザ表示幅の n% にする。
HSPACE = n	横方向のマージンを n ピクセルにする。
VSPACE = n	縦方向のマージンを n ピクセルにする。
BGCOLOR = color	フレームに色をつける。
ALIGN = TOP	マーキーに続くテキストを上部に表示。
ALIGN = MIDDLE	マーキーに続くテキストを中央に表示。
ALIGN = RIGHT	マーキーに続くテキストを下部に表示。
DIRECTION = LEFT	右から左へ流す。
DIRECTION = RIGHT	左から右へ流す。
LOOP = n	表示を n 回に限定する。
LOOP = INFINITE	無限に表示する。
BEHAVIOR = SCROLL	端から出て端に消える。
BEHAVIOR = SLIDE	端まで到達すると停止する。
BEHAVIOR = ALTERNATE	往復する。
SCROLLDELAY = n	移動時間間隔を n ミリ秒にする。
SCROLLMOUNT = n	移動距離間隔を n ミピクセルにする。

[演習]

画像を貼り付ける。

“絶対に優を取るぞ!!” と表示する。

JavaScriptでおみくじ

Java Scriptを用いて、おみくじページを作りましょう。たったのこれだけです。

List 3

omikuji.html

```
<HTML>

<HEAD>
<TITLE>
Java Scriptによるおみくじ
</TITLE>
<STYLE TYPE="text/css">
  A {text-decoration: none;}
  A:link {color: blue; text-decoration: none;}
  A:visited {color: blue; text-decoration: none;}
  A:active {color: red; text-decoration: none;}
  A:hover {color: red; background-color: lightgreen; }
  BODY { background-color: black; color: yellow; }
</STYLE>
</HEAD>

<BODY>

<DIV ALIGN = CENTER>
  今日の運勢は<BR><BR>

<SCRIPT LANGUAGE = "JavaScript">
// 0以上n未満の乱数を生成
function rand(n) {
  return Math.floor(n * Math.random());
}

var r = rand(6); // 運勢の種類の数

if (r == 0) { k = "大吉"; }
else if (r == 1) { k = "吉"; }
else if (r == 2) { k = "小吉"; }
else if (r == 3) { k = "末吉"; }
else if (r == 4) { k = "凶"; }
else if (r == 5) { k = "大凶"; }

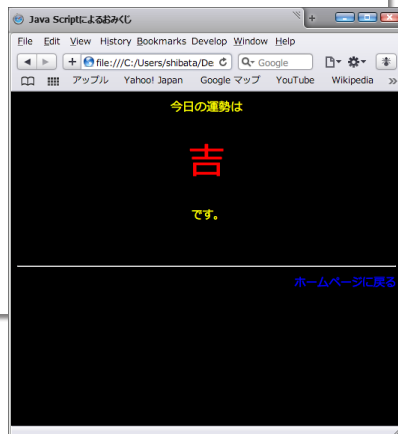
document.write("<FONT SIZE=7 COLOR=RED>" + k + "</FONT><BR>");
</SCRIPT>

<BR>です。<BR><BR><BR>
</DIV>

<HR>

<DIV ALIGN = RIGHT>
<A HREF = "index.html">ホームページに戻る</A>
</DIV>

</BODY>
</HTML>
```



■ JavaScript

JavaScript は、米 Netscape Communications が開発したスクリプト言語です。1996 年 2 月に発売した WWW ブラウザ Netscape Navigator 2.0 に実装されました。

言語仕様は Java に似ていますが、変数型や新規のクラス定義など、数多くの機能が省略されています。マウスのクリックなどの操作に合わせて Web ページの表示内容を動的に更新したり、フォームに入力されたデータのチェックをしたりするなど、HTML だけではできない機能を Web ページ上で実現できます。

実行速度が遅いという欠点がありますが、最新のブラウザでは実行速度の問題が改善されつつあります。

■ HTML への JavaScript の埋込み

JavaScript は、HTML 中に直通埋め込めるようになっています。<SCRIPT> タグを利用して、以下のように埋め込みます。

```
<SCRIPT LANGUAGE = "JavaScript">
  ... 中略 ...
</SCRIPT>
```

■ 関数

関数は、以下の形式で定義します。

```
function 関数名(引数) {
  ... 中略 ...
}
```

おみくじで定義している rand 関数は、0 以上 n 未満の乱数を生成して返却する関数です。

```
function rand(n) {
  return Math.floor(n * Math.random());
}
```

Math.random は 0.0 以上 1.0 未満の乱数を生成する関数であり、Math.floor(x) は x 以下の小数部を切り捨てた値を求める関数です。

- ▶ n が 6 であれば、0.0 以上 6.0 未満の乱数を生成して、その小数部を切り捨てた値を求めることとなりますので、結局 0 ~ 5 の乱数が得られます。

■ ドキュメントの更新

document.write(…) は、() の中に指示された内容をページに追加します。

表を作ろう

右に示すサイトは、ページ全体が『表』として作られています。すなわち、行と列とから構成されているのです。

- ▶ 各矩形部分（長方形の部分）は、Excelのセルに相当すると考えてよいでしょう。

このサイトのように、ページ全体が『表』としてレイアウトされているサイトは、数多く存在します。

ここでは、表について学習していくことにします。

まずは、基本的なタグの使い方を理解しましょう。

必ず理解しておかなければならないのが、<TABLE>、<TR>、<TH>、<TD>の4個のタグです。

<TABLE> … </TABLE>

表を作ります。

BORDER = n 外枠の太さをnピクセルにします。

<TR> … </TR>

表の行を指定します。

<TH> … </TH>

見出し項目を指定します。ブラウザによっても異なりますが、太字で表示されることが多いようです（そのため<TD>タグによる表示と区別がつきます）。

<TD> … </TD>

項目を指定します。

なお、TDタグとTHタグでは、表示文字をよせる方向を指定できます。

<TD ALIGN=CENTER> 中央に表示

<TD ALIGN=LEFT> 左側に表示

<TD ALIGN=RIGHT> 右側に表示



■ 演習：表の作成 (その1)

前ページで学習した基本的なタグを利用して、単純な表を表示してみましょう。

List 4

table01.html

```
<HTML>

<HEAD>
<TITLE>
オペレーティングシステム演習用ページ (表・その1)
</TITLE>
<STYLE TYPE="text/css">
  A {text-decoration: none;}
  A:link {color: blue; text-decoration: none;}
  A:visited {color: blue; text-decoration: none;}
  A:active {color: red; text-decoration: none;}
  A:hover {color: red; background-color: lightgreen; }
  BODY { background-color: white; color: black; }
</STYLE>
</HEAD>

<BODY>

<B>08A9999</B> <FONT COLOR=RED SIZE=5>柴田望洋</FONT>
<BR>
<BR>

<TABLE BORDER=10>
  <TR>
    <TH>プロセス名</TH>
    <TH>資源の占有順序</TH>
  </TR>

  <TR>
    <TH>プロセス A</TH>
    <TD ALIGN=CENTER> 1 → 2 → 3 </TD>
  </TR>

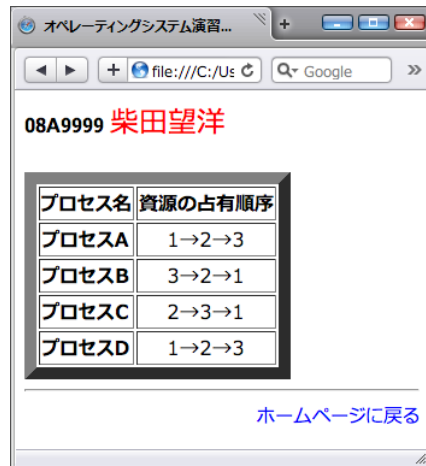
  <TR>
    <TH>プロセス B</TH>
    <TD ALIGN=CENTER> 3 → 2 → 1 </TD>
  </TR>

  <TR>
    <TH>プロセス C</TH>
    <TD ALIGN=CENTER> 2 → 3 → 1 </TD>
  </TR>

  <TR>
    <TH>プロセス D</TH>
    <TD ALIGN=CENTER> 1 → 2 → 3 </TD>
  </TR>
</TABLE>

<HR>
<DIV ALIGN = RIGHT>
<A HREF = "index.html">ホームページに戻る</A>
</DIV>

</BODY>
</HTML>
```



■ 代表的な画像ファイルの形式

代表的な画像ファイルの形式（JPEG / GIF / BMP / PNG）を紹介します。

◆ JPEG（Joint Photographic Experts Group：ジェイペグ）

静止画像のデジタルデータを圧縮する方式の一つです。なお、この方式を作った組織（ISO/IEC JTC 1/SC 29/WG 1）の略称を意味することもあります。拡張子としてはjpgまたはjpegが使われます。

非可逆圧縮の画像フォーマットとして知られていますが、可逆圧縮形式もサポートしています。ただし、可逆圧縮は特許などの関係でほとんど利用されていません。

◆ GIF（Graphics Interchange Format：ジフ／ギフ）

CompuServeが開発した画像ファイルフォーマットの一つです。LZW特許を使用した画像圧縮が可能です。拡張子はgifが使われます。

256色以下の画像を扱うことができる可逆圧縮形式です。圧縮画像ファイルフォーマットとして歴史が長く、WebブラウザではJPEGと並んで標準的にサポートされています。圧縮形式の特性上、同一色が連続する画像の圧縮率が高くなるため、イラストやボタン画像など、使用色数の少ない画像への使用に適しています。

◆ BMP（Microsoft Windows Bitmap Image：ビーエムピー）

マイクロソフトとIBMがWindowsとOS/2にわかれる前のOSを共同で開発していたころに作られた画像ファイル形式です。圧縮の方法についても定義されているものの、Windowsが標準では無圧縮のファイルを生成するため、他のアプリケーションにおいても、無指定時は、圧縮はされていない場合が多いようです。

◆ PNG（Portable Network Graphics：ピーエヌジー）

コンピュータ上で扱われる画像を格納するためのファイルフォーマットの一つです。既存の画像ファイルフォーマットであるGIFとTIFFの長所、すなわちインターネットでの流通（伝送）の効率、そしてより本格的な画像編集に耐える高度の表現力を兼ね備え、特にGIFを代替するフォーマットとしてインターネット上で独自に開発され、広く利用されています。

各タグには、豊富なオプションがあります。

`<TABLE> … </TABLE>`

- `BGCOLOR = color` 表の背景色を指定します。
- `BORDER = n` 外枠の太さを n ピクセルにします。
- `BORDERCOLOR = color` 外枠の色を指定します。
- `BORDERCOLORLIGHT = color` 外枠の明るい色を指定します。
- `BORDERCOLORDARK = color` 外枠の暗い色を指定します。
- `WIDTH = n` 横幅を n ピクセルにします。
- `WIDTH = n%` 横幅をブラウザ表示幅の $n\%$ にする。
- `CELLSPACING = n` セル間の間隔を n ピクセルにします。
- `CELLPADDING = n` セル内の余白を n ピクセルにします。

`<TR> … </TR>`

- `BGCOLOR = color` 行の背景色を指定します。

`<TH> … </TH>, <TD> … </TD>`

- `BGCOLOR = color` セルの背景色を指定します。
- `WIDTH = n` 横幅を n ピクセルにします。
- `WIDTH = n%` 横幅を表の $n\%$ にする。
- `ROWSAPN = n` セルを縦方向に n 行またがらせます。
- `COLSAPN = n` セルを横方向に n 列またがらせます。
- `ALIGN = CENTER` コンテンツを左右方向中央に表示
- `ALIGN = LEFT` コンテンツを左によせて表示
- `ALIGN = RIGHT` コンテンツを右側に表示
- `VALIGN = TOP` コンテンツを上によせて表示
- `VALIGN = BOTTOM` コンテンツを下によせて表示
- `VALIGN = MIDDLE` コンテンツを上下方向中央によせて表示
- `VALIGN = BASELINE` 最初の行を見やすい位置に表示

List 5

TableSample.html

```

<HTML>

<HEAD><TITLE>テーブルサンプル</TITLE></HEAD>

<BODY>

<TABLE WIDTH=100% BORDER=4 BORDERCOLORDARK=BLUE BORDERCOLORLIGHT=CYAN
        CELLSPACING=10 CELLPADDING=0 BGCOLOR=LIGHTYELLOW>
  <TR><TD WIDTH=50%>名前</TD>
    <TD WIDTH=50%>柴田望洋</TD>
  </TR>
</TABLE>

<BR><BR>

<TABLE WIDTH=100% BORDER=4 BORDERCOLORDARK=BLUE BORDERCOLORLIGHT=CYAN
        CELLSPACING=0 CELLPADDING=10>
  <TR><TD WIDTH=50%>名前</TD>
    <TD WIDTH=50% BGCOLOR=LIGHTGREEN>柴田望洋</TD>
  </TR>
</TABLE>

<BR><BR>

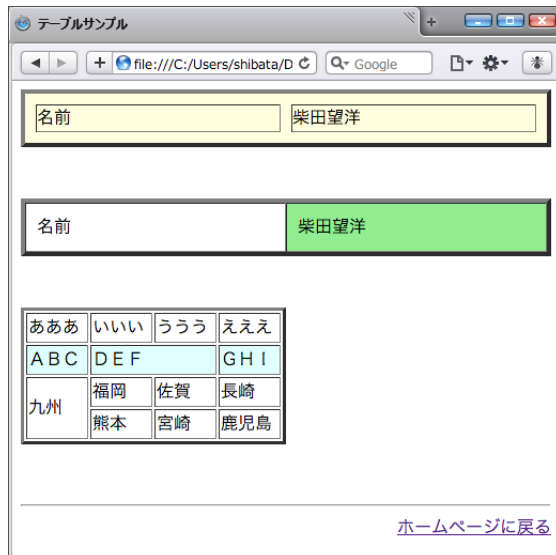
<TABLE BORDER=3 WIDTH=50%>
  <TR>
    <TD>あああ</TD>
    <TD>いいい</TD>
    <TD>ううう</TD>
    <TD>えええ</TD>
  </TR>
  <TR BGCOLOR=LIGHTCYAN>
    <TD>A B C</TD>
    <TD COLSPAN=2>D E F</TD>
    <TD>G H I</TD>
  </TR>
  <TR>
    <TD ROWSPAN=2>九州</TD>
    <TD>福岡</TD>
    <TD>佐賀</TD>
    <TD>長崎</TD>
  </TR>
  <TR>
    <TD>熊本</TD>
    <TD>宮崎</TD>
    <TD>鹿児島</TD>
  </TR>
</TABLE>

<BR><BR>

<HR>
<DIV ALIGN = RIGHT>
<A HREF = "index.html">ホームページに戻る</A>
</DIV>

</BODY>
</HTML>

```



List 6

table02.html

<HTML>

<HEAD>

<TITLE>

オペレーティングシステム演習用ページ (表・その2)

</TITLE>

<STYLE TYPE="text/css">

```
A {text-decoration: none;}
A:link {color: blue; text-decoration: none;}
A:visited {color: blue; text-decoration: none;}
A:active {color: red; text-decoration: none;}
A:hover {color: red; background-color: lightgreen;}
BODY { background-color: white; color: black; }
```

</STYLE>

</HEAD>

<BODY>

08A9999 柴田望洋

<TABLE BORDER=10>

<TR>

<TH ROWSPAN=2 BGCOLOR=YELLOW>プロセス名</TH>

<TH COLSPAN=3 BGCOLOR=CYAN> 資源の占有順序</TH>

</TR>

<TR>

<TH BGCOLOR=GREEN> 資源 X </TH>

<TH BGCOLOR=BLUE> 資源 Y </TH>

<TH BGCOLOR=RED> 資源 Z </TH>

</TR>

<TR>

<TH>プロセス A</TH>

<TD ALIGN=CENTER> 1 </TD><TD ALIGN=CENTER> 2 </TD><TD ALIGN=CENTER> 3 </TD>

</TR>

<TR>

<TH>プロセス B</TH>

<TD ALIGN=CENTER> 3 </TD><TD ALIGN=CENTER> 2 </TD><TD ALIGN=CENTER> 1 </TD>

</TR>

<TR>

<TH>プロセス C</TH>

<TD ALIGN=CENTER> 2 </TD><TD ALIGN=CENTER> 3 </TD><TD ALIGN=CENTER> 1 </TD>

</TR>

<TR>

<TH>プロセス D</TH>

<TD ALIGN=CENTER> 1 </TD><TD ALIGN=CENTER> 2 </TD><TD ALIGN=CENTER> 3 </TD>

</TR>

</TABLE>

<HR>

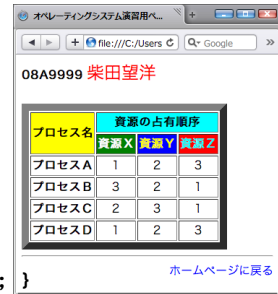
<DIV ALIGN = RIGHT>

ホームページに戻る

</DIV>

</BODY>

</HTML>



■ フォントとレンダリングについて確認する

Windows におけるフォントと代替フォント・アンチエイリアスの関係を理解する。以下に示すのは、MS ゴシック用のものです。これを参考にして、これ以外のフォントについて、同等な表を作成します。

List 7

TableSample.html

```
<HTML>

<HEAD>
<TITLE>
オペレーティングシステム演習用ページ (MS ゴシックのアンチエイリアス)
</TITLE>
<STYLE TYPE="text/css">
  A {text-decoration: none;}
  A:link {color: blue; text-decoration: none;}
  A:visited {color: blue; text-decoration: none;}
  A:active {color: red; text-decoration: none;}
  A:hover {color: red; background-color: lightgreen; }
  BODY { background-color: white; color: black; }
</STYLE>
</HEAD>

<BODY>

<B>08A9999</B> <FONT COLOR=RED SIZE=5>柴田望洋</FONT><BR><BR>

『MS ゴシック』フォントのサイズと、ビットマップフォントによる代替表示・アンチエイリアス (ClearTypeによるスムージング) の対応を調べてみます。<BR>
<BR>

<TABLE BORDER=1>
  <TR>
    <TH>ポイント</TH>
    <TH>ビットマップ</TH>
    <TH>アンチエイリアス</TH>
    <TH>画像</TH>
  </TR>

  <TR>
    <TD ALIGN=CENTER>12
    <TD ALIGN=CENTER ROWSPAN=3>○
    <TD ALIGN=CENTER ROWSPAN=3>×
    <TD><IMG SRC="12.png">
  </TR>

  <TR>
    <TD ALIGN=CENTER>14
    <TD><IMG SRC="14.png">
  </TR>

  <TR>
    <TD ALIGN=CENTER>16
    <TD><IMG SRC="16.png">
  </TR>

  <TR>
    <TD ALIGN=CENTER>18
```

```

<TD ALIGN=CENTER ROWSPAN=4>×
<TD ALIGN=CENTER ROWSPAN=4>○
<TD><IMG SRC="18.png">
</TR>

<TR>
<TD ALIGN=CENTER>20
<TD><IMG SRC="20.png">
</TR>

<TR>
<TD ALIGN=CENTER>22
<TD><IMG SRC="22.png">
</TR>

<TR>
<TD ALIGN=CENTER>24
<TD><IMG SRC="24.png">
</TR>

```

</TABLE>

<HR>

<DIV ALIGN = RIGHT>

ホームページに戻る

</DIV>

</BODY>

</HTML>



■ javadoc による HTML 作成

Java では、ソースプログラムに埋め込んだコメントから、そのプログラムのマニュアルともいえるべき文書を HTML 形式で作成できます。

■ 文書化コメント

Java の文書化コメントは、`/**` と `*/` で囲んだコメントです。コメントが複数行にわたるときは、以下に示すように、中間行の先頭にも `*` を書くのが一般的です。

中間行の先頭に位置する `*` と、それより左側の空白とタブは、読み捨てられます。

文書化コメントは、クラス・インタフェース・コンストラクタ・メソッド・フィールドの宣言の直前に置かれているときにだけ認識されます。

コメントには HTML のタグをそのまま使うことができます。たとえば、`` と `` で囲んだ部分は太字となり、`<I>` と `</I>` で囲んだ部分はイタリック体となります。また、`
` と書いた箇所で行改めされることとなります。

*

コメントの最初には《主説明》を記述します。この部分には複数の文を記述できますが、先頭の文は、コメントの対象となるクラスやメソッドなどの《概要》を簡潔にまとめた文書でなければなりません。

- ▶ 先頭の文だけが、《概要》として抽出されます。

主説明の後ろには、プログラムの著者名・メソッドの戻り値などを、文書化コメント専用のタグを用いて記述します。タグは `@` で始まる特別な命令です。

@author “著者名”

ドキュメントに《著者》の項目を追加して、“著者名”を書き込みます。

- ▶ 一つの `@author` に複数の“著者名”を書くこともできますし、個々の“著者名”ごとに `@author` を与えることもできます。

{@code “コード” }

プログラムコードであることを示します。

- ▶ 生成された HTML では、`<code>` タグと `</code>` タグで囲まれます。そのため、ほとんどのブラウザで等幅フォントで表示されます。プログラム部分や変数名などを示すときに利用します。

@return “戻り値”

ドキュメントに《戻り値》の項目を追加して、“戻り値”を書き込みます。メソッドの戻り値の型や値に関する情報を記述します。メソッドのコメントでのみ有効です。

@param “引数名” “解説”

ドキュメントに《パラメータ》の項目を追加して“引数名”とその“解説”を書き込みます。メソッド・コンストラクタ・クラスのコメントでのみ有効です。

- ▶ パラメータとは、メソッドの仮引数のことです。

@see “参照先”

ドキュメントに《関連項目》の項目を追加して“参照先”を指すリンクあるいは文書を書き込みます。@see タグの個数は任意です。

このタグには、以下の3種類の形式があります。

- @see "文字列"

"文字列"を追加します。リンクは生成されません。URLではアクセスできない情報の参照先を示す場合に利用します。

- @see label

URL#value で定義されているリンクを追加します。URL#value は、相対URLまたは絶対URLです。

- @see package.class#member label

指定された名前をもつメンバに関するドキュメントを指すリンクを、表示テキスト label とともに追加します。label は省略可能です。label を省略すると、リンク先のメンバの名前が適切に短縮されて表示されます。

■ javadoc ツール

ソースプログラム中に記入された文書化コメントをもとに、ドキュメントを作成するのが javadoc ツールです。このツールの起動は以下のように行います。

```
javac -Xdoclint:none -d docs -s docs -Xdoclint:none -Xdoclint:none -Xdoclint:none @オプション パッケージ名 ソースファイル @引数ファイル
```



←このようなHTML
が自動生成されます。

■ インラインフレームによる外部ファイルの表示

インラインフレームを利用して外部ファイルを表示してみましょう。

- ▶ プリント “オペレーティングシステム” の pp.52 ~ 54 の各ソースプログラムが WWW/OS/ Thread ディレクトリに格納されていなければなりません。

List 8

Sleep.html

```
<HTML>
<HEAD>
<TITLE>Sleep. java</TITLE>
</HEAD>

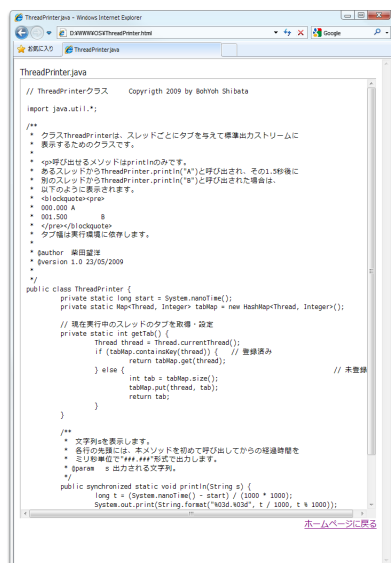
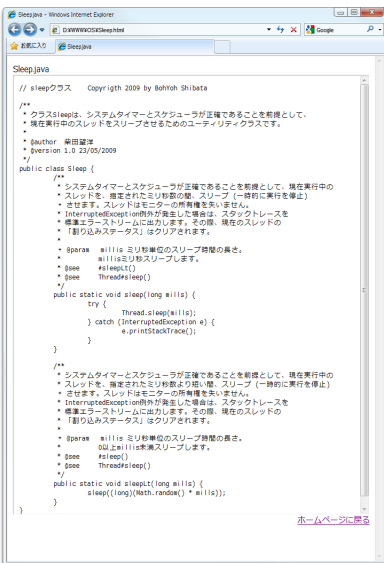
<BODY>
Sleep. java
<IFRAME WIDTH=100% HEIGHT=90% SRC="Thread/Sleep. java">
</IFRAME>
<DIV ALIGN=RIGHT><A HREF="index.html">ホームページに戻る</A></DIV>
</BODY>
</HTML>
```

List 9

ThreadPrinter.html

```
<HTML>
<HEAD>
<TITLE>ThreadPrinter. java</TITLE>
</HEAD>

<BODY>
ThreadPrinter. java
<IFRAME WIDTH=100% HEIGHT=90% SRC="Thread/ThreadPrinter. java">
</IFRAME>
<DIV ALIGN=RIGHT><A HREF="index.html">ホームページに戻る</A></DIV>
</BODY>
</HTML>
```



List 10

Sleep.html

```

<HTML>
<HEAD>
<TITLE>MyThreadTester.java</TITLE>
</HEAD>

<BODY>
MyThreadTester.java
<IFRAME WIDTH=100% HEIGHT=60% SRC="Thread/MyThreadTester.java">
</IFRAME>
<BR>

<TABLE WIDTH=100% BORDER=0>
<TR><TD WIDTH=50%>synchronizedなし<TD WIDTH=50%>synchronizedあり</TD></TR>
</TABLE>
<NOBR>
<IFRAME WIDTH=50% HEIGHT=30% SRC="Thread/NONSYNCHRONIZED.TXT">
</IFRAME>
<IFRAME WIDTH=50% HEIGHT=30% SRC="Thread/SYNCHRONIZED.TXT">
</IFRAME>
</NOBR>

<DIV ALIGN=RIGHT><A HREF="index.html">ホームページに戻る</A></DIV>
</BODY>
</HTML>

```

```

MyThreadTester.java
// 二つのスレッド (各スレッドは変数xの値を取り出して1を加える)
class MyThread implements Runnable {
    static int x;

    public synchronized void go() {
        int temp = x;
        ThreadPrinter.println("++" + temp);
        x = temp + 1;
        ThreadPrinter.println("--" + x);
    }

    public void run() {
        for (int i = 0; i < 5; i++) {
            go();
        }
    }
}

public class MyThreadTester {
    public static void main(String[] args) {
        MyThread job = new MyThread();
        Thread a = new Thread(job); // スレッドaの作成
        Thread b = new Thread(job); // スレッドbの作成
        a.start(); // スレッドaの実行開始
        b.start(); // スレッドbの実行開始
    }
}

synchronizedなし      synchronizedあり
000.928 -->1          000.627 -->3
000.934 -->2          000.627 -->4
000.939 -->2          000.627 -->4
001.134 -->2          001.027 -->5
001.135 -->2          001.027 -->5
001.334 -->3          001.227 -->6
001.335 -->3          001.227 -->6
001.534 -->3          001.428 -->7
001.535 -->3          001.428 -->7
001.735 -->4          001.628 -->8
001.735 -->4          001.628 -->8
001.935 -->4          001.829 -->9
001.935 -->4          001.829 -->9
002.135 -->5          002.030 -->10
002.135 -->5

```

エディタでMyThreadTester.javaのsynchronizedキーワードを消す。

ドライブHのWWW/OS/Thread上で以下の作業を行って作成。

```
javac MyThreadTester.java
```

```
java MyThreadTester > NONSYNCHRONIZED.TXT
```

エディタでMyThreadTester.javaのsynchronizedキーワードを付ける。

ドライブHのWWW/OS/Thread上で以下の作業を行って作成。

```
javac MyThreadTester.java
```

```
java MyThreadTester > SYNCHRONIZED.TXT
```


JavaScript による時間の取扱い

みなさんは、留年しなければ2012年3月20日に卒業することになります。卒業までの日数を計算して表示しましょう。

「おみくじ」と同様に、JavaScript を利用します。

おみくじでは `Math.floor()` というメソッドを利用しましたが、今回は `Math.ceil()` というメソッドを利用します。これらのメソッドの概略は、以下のとおりです。

□ `Math.floor(x)`

実数値 x 以下の最大の整数を求めて、小数部を切り捨てる。

□ `Math.ceil(x)`

実数値 x 以上の最小の整数を求めて、小数部を切り上げる。

それでは、トップページ (`index.html`) のボディ部の適当な位置に以下の部分を追加してみましょう。

List 11

`index.html`

```
<script language="JavaScript">
<!--

var today = new Date();           // 現在
var graduation = new Date(2012, 2, 20); // 卒業式 (年, 月-1, 日)
diff = Math.ceil((graduation.getTime() - today.getTime()) / (24*60*60*1000));
document.write("あと", diff, "日で卒業です。");

-->
</script>
```

日付を指定して `Date` クラスのオブジェクト (変数) を作る際は、月の値を1だけ減らさなければなりません。すなわち、1月、2月、…、12月は、それぞれ0, 1, …, 11です。そのため、2012年3月20日を表すのは、`Date(2012, 2, 20)` となるのです。

- ▶ JavaScript だけでなく、C言語やJavaの日付関連のライブラリも、1～12月を0～11の値で表し、1～31日を1～31の値で表します。すなわち、『月』は0から始まる値で、『日』は1から始まる値です。

このようなアンバランスな理由は、以下のとおりです。

- ・コンピュータの内部の計算は1から始まる値よりも0から始まる値のほうが都合がよい。
- ・英語圏では、月は“January”, “February”, …と表すため、“1月”, “2月” …と、月を数値で表すという発想そのものがない。

現在の時刻と、卒業式の時刻 (2012年3月20日0時0分0秒) との差を求めます。求めた値はミリ秒単位ですから、日単位に変換します。

【課題】 次回の誕生日までの日数を表示してみよう。

■ インタバルタイマ

次に、ホームページに訪訪してからの秒数を表示することになります。前ページと同様に、トップページ (index.html) のボディ部の適当な位置に以下の部分を追加してみましょう。

List 12

index.html

```
このページに来てから<SPAN ID="ELASPE">0</SPAN>秒です。 <BR>

<script language="JavaScript">
<!--

var begin = new Date();      // 来訪時

function Time(){
    crnt = new Date();      // 現在
    document.getElementById("ELASPE").innerHTML =
        Math.ceil((crnt.getTime() - begin.getTime()) / 1000);
}

setInterval("Time()", 1000);

-->
</script>
```

ここでは、DHTML (Dynamic HTML / 動的 HTML) を利用しています。DHTMLとは、静的な HTML の内容を動的に変更するウェブ技術を指す抽象概念のことです。CSS や JavaScript 等のクライアントサイドスクリプト言語を用いることによって、動き(変化)のある対話的なページを小容量で実現できます。

*

HTML では、いったん『このホームページに来てから 0 秒です。』と表示しておきます。この中の“0”の部分は、SPAN タグで囲まれ、ELAPSE という名前が与えられています。

setInterval 関数によって、Time 関数を 1000 ミリ秒 = 1 秒ごとに呼び出すようにインタバルタイマをセットします。

1 秒ごとに呼び出される Time 関数は、現在の時刻 crnt と、ホームページロード時の時刻 begin との差を秒単位の値として求めます。そして、ドキュメント中の ELAPSE という前の部分を、求めた秒数と入れ替えます。

■ スタッフロールを作る

映画の最後に、監督や出演者などの字幕がスクロールされて表示されますね。あれを作りましょう。

List 13

staff1.html

```
<HTML>
<HEAD>
<TITLE>
  柴田望洋のオペレーティングシステム専用ホームページについて
</TITLE>
<STYLE TYPE="text/css">
  A {text-decoration: none;}
  A:link {color: yellow; text-decoration: none;}
  A:visited {color: yellow; text-decoration: none;}
  A:active {color: red; text-decoration: none;}
  A:hover {color: red; background-color: lightgreen; }
  BODY { background-color: black; color: white; }
</STYLE>
</HEAD>

<BODY>

<DIV ALIGN=CENTER>
<FONT SIZE=5 COLOR=YELLOW>柴田望洋のオペレーティングシステム専用ホームページに
ついて</FONT><BR>
</DIV>

<HR>
<BR>

<MARQUEE DIRECTION=UP HEIGHT=70% SCROLLAMOUNT=2>

<DIV ALIGN=CENTER>
Producer<BR>
<FONT SIZE=5 COLOR=YELLOW>柴田望洋</FONT><BR>
<BR>

Copyright<BR>
<FONT SIZE=5 COLOR=YELLOW>柴田望洋</FONT><BR>
<BR>

作成開始日<BR>
<FONT SIZE=5 COLOR=YELLOW>200X年4月1日</FONT><BR>
</DIV>

</MARQUEE>

<HR>
<DIV ALIGN=RIGHT>
<A HREF="index.html">ホームページへ戻る</A>
</DIV>
<HR>

</BODY>
</HTML>
```

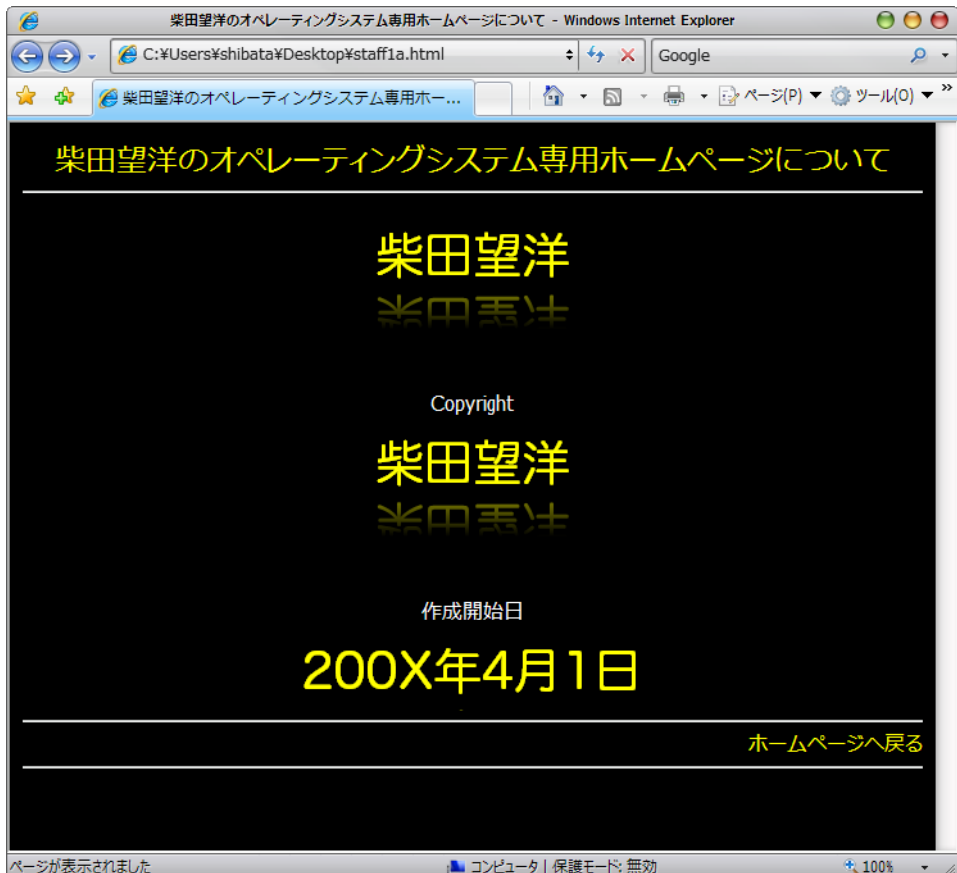
プロデューサ、著作権者、作成開始日以外の項目を追加しよう !!

単にテキスト (文字) を表示するだけでは面白くありません。ここでは、文字に対して反射効果 (水面に反射しているような効果) を付けた画像を作って、その画像を張り込むことにします。

文字やその反射部分は背景に溶け込まなければなりませんから、透過効果をサポートした画像形式である png を使うことにします。

以下のような画像ファイルを作成して、その画像を表示させましょう。もちろん、図の黒い部分は、「黒」として作成するのではなく、「透明」として作成しなければなりません。

画像ファイルは、情報処理センターにインストールされているソフトウェアだけで簡単に作成できます。



BMI を計算する

右のようなページを作りましょう。身長と体重を入力し、「計算」ボタンを押すと、BMIが表示されます。

List 14

BMI.html

```
<HTML>
<HEAD>
<TITLE>
  BMIを計算しよう
</TITLE>
<STYLE TYPE="text/css">
  A {text-decoration: none;}
  A:link {color: blue; text-decoration: none;}
  A:visited {color: blue; text-decoration: none;}
  A:active {color: red; text-decoration: none;}
  A:hover {color: red; background-color: lightgreen; }
  BODY { background-color: white; color: black; }
</STYLE>
</HEAD>

<BODY>

<P>BMIは<B>体重 (kg)÷身長(m)<SUP>2</SUP></B>によって求められる値です。BMIが22
程度が病気になりにくいという統計があります。</P>

<SCRIPT LANGUAGE = "JavaScript">
<!--
  function CalcBMI(){
    var Height = document.BMI.Sincho.value;
    var Weight = document.BMI.Taijyu.value;
    var CM = (Height / 100);
    var Value = (Weight / (CM * CM)) + "0";
    document.getElementById("BMI_VALUE").innerHTML = Value.substring(0, 5);
  }
-->
</SCRIPT>

<FORM name = "BMI">
  身長 : <input type = "text" size = "6" name = "Sincho">cm<BR><BR>
  体重 : <input type = "text" size = "6" name = "Taijyu">kg<BR><BR>
  <input type = "button" value = "計算" onclick= "CalcBMI()"><BR><BR>
</FORM>

<P>BMIは<SPAN ID="BMI_VALUE">0</SPAN>です。</P>

<HR>
<DIV ALIGN = RIGHT>
<A HREF="index.html">戻る</A>
</DIV>

</BODY>
</HTML>
```

BMI について

BMIはBody Mass Indexの略であり、体重（kg）を身長（m）の二乗で割った値です。CTスキャナ法で判定したのと類似した相関関係があり、合併症発症率との間でも確認され、肥満度の指標としての地位を確立しつつあります。

肥満の判定

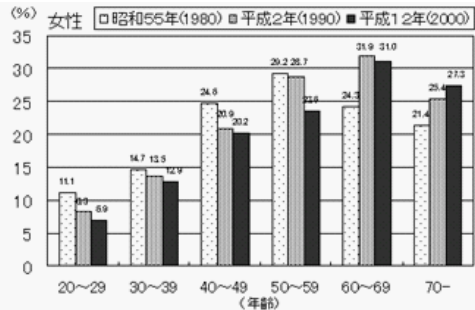
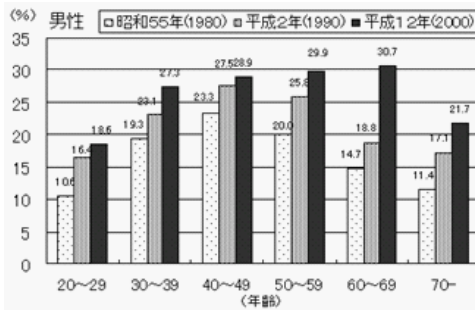
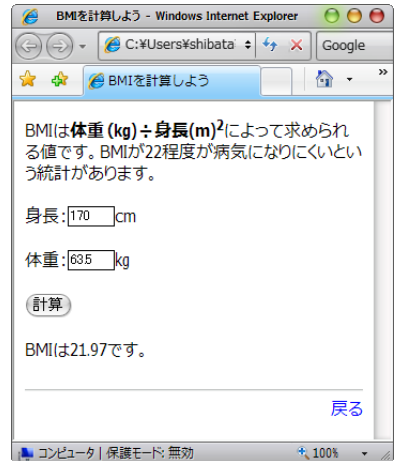
BMI = 22 を標準とします。BMIは「 $\text{体重} / \text{身長}^2$ 」により算定します。体重の単位はkgで、身長の単位はmです。

BMI	18.5未満	18.5以上25未満	25以上
判定	やせ	正常	肥満

（日本肥満学会 肥満の判定基準より）

厚生労働省『平成12年度国民栄養調査』によると、特に男性の肥満者が年々増加していることが分かります（下のグラフはBMIが25以上の者の割合）。

30歳代～60歳代の男性の肥満者は、ほぼ30%であり、軽度な肥満を入れると、日本人男性の約半分は肥満ということになります。



■ 訪問回数を表示する

自分のホームページを見ている人が、何回目に訪れたのかを表示しましょう。index.htmlの<BODY>部の先頭に、以下のものを追加してください。

List 15

index.html

```
<script language = "JavaScript">
<!--

function setCount(n) {
    theDay = 30;
    setDay = new Date();
    setDay.setTime(setDay.getTime() + (theDay * 1000 * 60 * 60 * 24));
    expDay = setDay.toGMTString();
    document.cookie = "count=" + n + ";expires=" + expDay;
}

function getCount() {
    theName = "count=";
    theCookie = document.cookie + ";";
    start = theCookie.indexOf(theName);
    if (start != -1) {
        end = theCookie.indexOf(";", start);
        count = eval(unescape(theCookie.substring(start+theName.length,end)));
        document.writeln(count + "回目のアクセスですね。");
        setCount(count+1);
    } else {
        document.writeln("初めてのアクセスありがとうございます。");
        setCount(2);
    }
}

getCount();

-->
</script>
```

クッキーとは、Webサイトを訪れたユーザを識別するために、WWWサーバがWWWブラウザに埋め込む少容量のユーザ情報のことです。WWWサーバは、クッキーによってユーザ（厳密にはWWWブラウザ）を識別し、そのユーザに適したWeb情報を提供するパーソナライズを行ったり、電子商取引におけるセッション管理にクッキーの仕組みを利用したりします。

パーソナライズを行う目的は、顧客の特性に適合した情報を提供することで顧客にとって魅力のある情報を提供し、競合他社のWebサイトとの差別化を図ったり、ワンツーワンマーケティングによる提供情報の最適化によって顧客の固定化をねらうことなどです。

最終更新日時を表示する

ホームページの最終更新の日付・時間を表示しましょう。index.htmlの<BODY>部の適当な箇所に、以下のものを追加します。

List 16

index.html

```
<script language = "JavaScript">
<!--
document.writeln("最終更新：" + document.lastModified);
-->
</script>
```

表示は、『最終更新：11/18/2009 03:12:58』といった形式です。
日本語で表示したければ、以下のようしましょう。

List 17

index.html

```
<script language = "JavaScript">
<!--
wd = new Array("日", "月", "火", "水", "木", "金", "土");
mod = new Date(document.lastModified);
document.writeln("最終更新："
    + mod.getFullYear()      + "年"
    + (mod.getMonth() + 1)   + "月"
    + mod.getDate()          + "日"
    + "(" + wd[mod.getDay()] + ") "
    + mod.getHours()         + "時"
    + mod.getMinutes()       + "分"
    + mod.getSeconds()       + "秒"
    );
-->
</script>
```

こうすると、『最終更新：2009年11月18日(水)03時12分58秒』といった形式で表示されます。

▶ 日時だけでなく、曜日も表示されます。

最終更新日時が表示されていないページの最終更新日時を調べるのは簡単です。ブラウザのアドレス欄に javascript:alert(document.lastModified) と書くだけです。

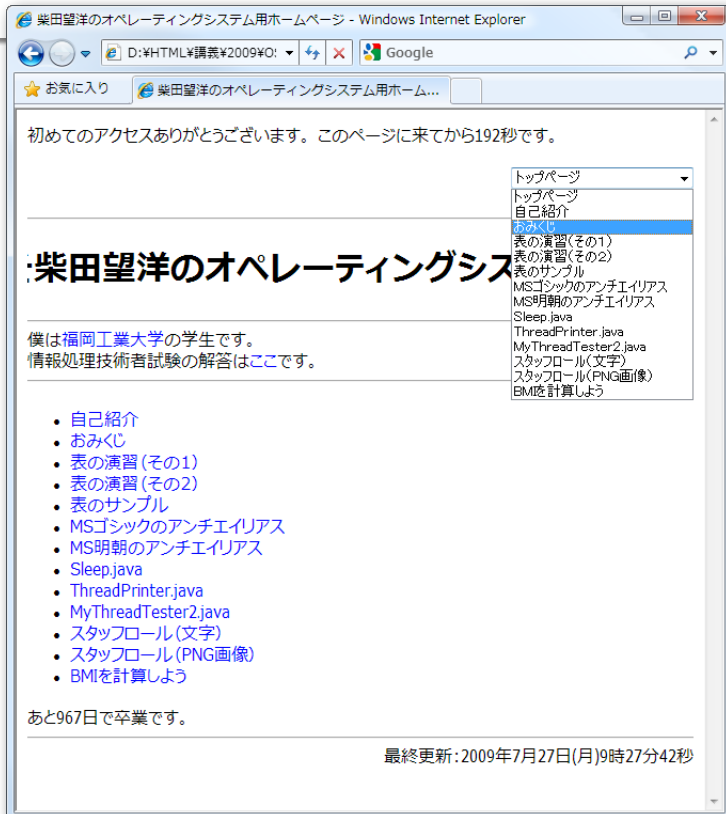
プルダウンによるナビゲーション

自分のサイト内の別ページへのリンクを、プルダウンメニューによって選択できるようにします。

List 18

index.html

```
<DIV ALIGN =RIGHT>
<FORM>
<SELECT onChange="location=this.options[this.selectedIndex].value">
  <OPTION SELECTED VALUE="index.html">トップページ
  <OPTION VALUE="introduction.html">自己紹介
  <OPTION VALUE="omikuji.html">おみくじ
  <OPTION VALUE="table01.html">表の演習 (その1)
  <OPTION VALUE="table02.html">表の演習 (その2)
  <OPTION VALUE="TableSample.html">表のサンプル
  <OPTION VALUE="MSG/MSGothic.html">MSゴシックのアンチエイリアス
  <OPTION VALUE="MSM/MSMincho.html">MS明朝のアンチエイリアス
  <OPTION VALUE="Sleep2.html">Sleep.java
  <OPTION VALUE="ThreadPrinter2.html">ThreadPrinter.java
  <OPTION VALUE="MyThreadTester2.html">MyThreadTester2.java
  <OPTION VALUE="staff1.html">スタッフロール (文字)
  <OPTION VALUE="staff2.html">スタッフロール (PNG画像)
  <OPTION VALUE="BMI.html">BMIを計算しよう
</SELECT>
</FORM>
</DIV>
```



まず、<FORM>と</FORM>によって、FORMを作成します。ここでは、メニューをページの右端に表示させるため、<DIV ALIGN=RIGHT>と</DIV>で囲んでいます。

FORMの中には、プルダウンメニューのためにSELECTタグを挿入します。

SELECTの中には、プルダウンメニューで表示するための項目を列挙します。

各項目はOPTIONタグによって記述します。SELECTED属性は、既定で選択される項目に指定します。VALUE属性には、選択されたときのジャンプ先のURLを指定します。

ジャンプは、SELECT中のonChangeによって指定します。これで、メニューが選択されると、以下のJavaScriptが実行されることになります。

```
location=this.options[this.selectedIndex].value
```

locationに代入するのは、ブラウザに読み込ませるべきページのURLです。

this.optionは、SELECTタグ内のOPTIONタグの配列を表します。また、選択された項目の番号（添字）はthis.selectedIndexに入ります。そのため、this.option[this.selectedIndex].valueが選択された項目のURLとなります。

参考文献

- 1) 大藤幹『詳解HTML&XHTML&CSS辞典』, 秀和システム, 2002
- 2) W3C / 内田明ら (翻訳) HTML 4.01 仕様書,
<http://www.asahi-net.or.jp/~SD5A-UCD/rec-html401j/cover.html>